Asian Journal of Research in Computer Science

Volume 16, Issue 4, Page 327-335, 2023; Article no.AJRCOS.109404
ISSN: 2581-8260

Language Agnostic Ontology Extension Framework

Reva Bharara [a*], **Aryan Rathore** [a] **and Krishna Kumar Tiwari** [b]

[a] *Vellore Institute of Technology, Bhopal, India.*
[b] *Head of Knowledge Graph Platform AI Center of Excellence, Jio Private Limited Bangalore, India.*

Authors' contributions

This work was carried out in collaboration among all authors. All authors read and approved the final manuscript.

Article Information

DOI: 10.9734/AJRCOS/2023/v16i4394

Open Peer Review History:

This journal follows the Advanced Open Peer Review policy. Identity of the Reviewers, Editor(s) and additional Reviewers, peer review comments, different versions of the manuscript, comments of the editors, etc are available here:Complete Peer review History: https://www.sdiarticle5.com/review-history/109404

Received: 20/09/2023
Accepted: 24/11/2023
Published: 11/12/2023

Original Research Article

ABSTRACT

Ontologies are powerful structures used to define the schema and organization of knowledge. They provide a framework for precisely and explicitly organizing concepts, entities, attributes, and relationships within a domain, enabling effective data management, knowledge sharing, and intelligent decisionmaking. Knowledge graphs, on the other hand, store data in a graphical form, facilitating semantically rich and interconnected data analysis, management, and exploration. Nodes represent entities, edges depict relationships between nodes, and attributes showcase node properties. Combining the strengths of ontology, which offers schema and concept-level modeling, with the data richness of knowledge graphs, one can unlock powerful reasoning and inference capabilities that closely resemble facts and truths. The symbiotic relationship between knowledge graphs and ontology, wherein ontology serves as a blueprint for representing knowledge in the graphs, creates a robust foundation for knowledge representation. Traditionally, ontologies have been created by experts or skilled teams with domain knowledge, aiming to build comprehensive ontologies supporting holistic data representation. However, these domain-specific ontologies require manual intervention for updates and remain language-specific, making it challenging to transfer them across different language formats. To address this

*Corresponding author: E-mail: bhararareva@gmail.com;

challenge, a crucial component is an automatic generation framework for language-independent ontologies. Such a capability would provide a data-driven approach for creating necessary ontologies. The process involves taking a text corpus as input and subsequently constructing a knowledge graph through named entity recognition. This graph is then transformed into a concept-level modeling graph, where an ID-based approach is utilized to represent the concepts. The ID-based concepts can be extended to facilitate merging and updating the ontology with new relevant information. Concurrently, this approach can also be expanded to achieve language independence, enabling the conversion of the ontology to any required language. Expressing these ontologies in mathematical terms is essential to achieve language-agnosticism, ensuring completeness, relevance, and independence from specific domains, thereby making them applicable across various linguistic contexts.

Keywords: Ontology; knowledge graphs; named entity recognition; RDF; language-agnostic.

1 INTRODUCTION

Ontologies have proven to be an effective and robust schema framework for knowledge representation [1]. This has, in turn, influenced the schema of knowledge graphs, providing them with a powerful blueprint that specifies entities and their relations [1]. Knowledge graphs, as graphical data structures, excel in storing and representing information through interconnected nodes and edges, effectively representing real-world entities and their relationships [2]. The intricate interconnections offered by knowledge graphs enable efficient data analysis and facilitate the discovery of previously unknown relationships, patterns, and insights. The data stored in knowledge graphs can originate from structured data sources such as datasets or can be extracted from unstructured text. The latter necessitates a considerable amount of preprocessing and natural language processing to extract knowledge. In the scope of this paper, we concentrate on the generation of knowledge graphs from raw text corpora.

Raw text data is diverse and consists of unorganized strings of characters, lacking discernible structure or semantic meaning. Therefore, to extract meaningful information from raw text input, advanced natural language processing techniques and intelligent knowledge acquisition methodologies are essential [3]. The foundational structure of knowledge graphs relies on triples, which comprise subject-predicate-object entities and form the fundamental unit in knowledge graph construction [4]. By transforming text input into these triples through the use of pretrained pipelines and large language models, and performing named-entity recognition, one can establish the groundwork for supporting advanced reasoning, inference, and data exploration [5]. This, in turn, aids in the process of ontology generation.

Once the knowledge graph representation is complete, the next steps involve labeling the entities representing subjects and objects [6]. The objective is to generalize the nodes and extract essential relations for concept-level modeling of the data, leading to ontology generation. This process can be facilitated through mechanisms like text annotation, which allows for a data-driven approach to ontology generation with human intelligence intervention [7].

The resulting ontology, however, remains language-specific. To achieve language-agnosticism, the next step involves representing the entities and relationships in the ontology using specific mathematical representations [8]. These mathematical representations are derived from the core ontology generated in the language of the input text corpus, essentially capturing the concepts in a mathematical form [9]. Subsequently, these mathematical representations are mapped to their corresponding meanings in different languages using various translation APIs like Google Translate. As a result, the base ontology is stored in a mathematical format, containing entities and relations represented in mathematical representations [9]. This format allows for transformation into any desired language, making the ontology language agnostic.

2 PREREQUISITE

2.1 Input Data

For the creation of a knowledge graph from the data, which marks the initial step of the automatic ontology generation process, a collection of 50 text corpora was utilized. Each text corpus contained details about fictitious individuals and companies, randomly generated, and encompassed information such as their names, occupations, personality traits, and more. The input data was provided in raw text format, representing unstructured data, which was then processed using natural language processing techniques to extract crucial and meaningful information.

2.2 Named Entity Recognition and Relation Extraction

The input text corpora are processed to extract entities using large language models that demonstrate excellent performance in entity extraction tasks due to their ability to understand contextual information and language patterns. This is accomplished through a mechanism called named-entity recognition (NER). The process commences with text segmentation into tokens, such as words or subwords, and the removal of stopwords (commonly used words like ”is,” ”that”). The tokens then undergo part-of-speech (POS) tagging, which labels them according to grammatical parts of speech, such as nouns, pronouns, verbs, adverbs, adjectives, etc. Subsequently, the tokens are analyzed using NER models to identify entities and extract relations. These entities and relations are then converted to a subject-predicate-object format, allowing them to be represented in the form of a data-level knowledge graph.

2.3 Knowledge Graph Creation

The data, now extracted in triples format, is imported into an automatic knowledge graph generation database, such as ArangoDB. In this database, the triples are connected to form a graph, where nodes represent the subjects and objects, while edges depict the relationships between these nodes. The resulting knowledge graph comprises interconnected details of users and comapanies as depicted in Fig. 1 and Fig. 2, enabling a semantic web that facilitates data exploration and the discovery of hidden links and insights. This interconnected structure allows for a deeper understanding of the data and enhances the potential to uncover valuable connections within the information.

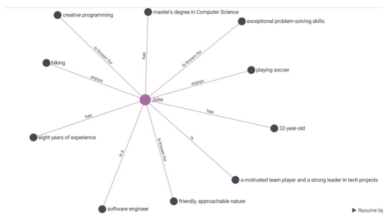

Fig. 1. Knowledge graph generated for sample text 1

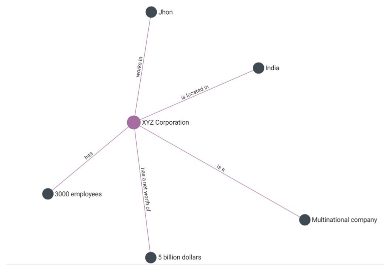

Fig. 2. Knowledge Graph generated for sample text 2

3 METHODOLOGY

3.1 Approach

3.1.1 Ontology creation

The knowledge graph derived from the text corpus requires some cleaning to eliminate unwanted and irrelevant entities. This process involves identifying nodes that lack connections to any other nodes. Once this cleaning is completed, the nodes undergo annotation using NER models to categorize them into concepts. For instance, the name ”John” is annotated as ’Person’, and ”32-year-old” is categorized as ’age’, while maintaining the original relationships represented by the edges. To perform this annotation, we employed large language models, as well as pretrained pipelines available in libraries like SpaCy and other NLP frameworks. These models classify text tokens and entities into broad categories, such as person, organization, date, time, and more. They have been trained on diverse data from sources like news text, web texts, literature, and other domains, ensuring accurate annotations.

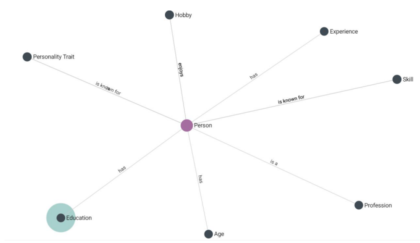

Fig. 3. Ontology generated for sample text 1

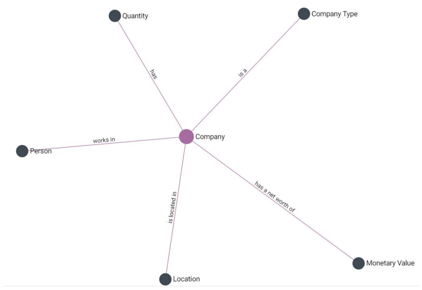

Fig. 4. Ontology generated for sample text 2

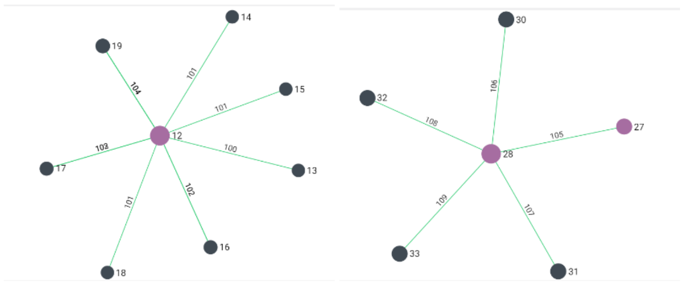

Fig. 5. Language agnostic ontology for sample 1 and sample 2 respectively

3.1.2 Language agnostic ontology generation

The ontology created so far is specific to the language of the text corpus and, as a result, lacks language independence. To achieve language agnosticism, we can represent the ontology in a mathematical format that can be easily translated into different languages. Each mathematical representation corresponds to a concept, which can be converted into various languages using a translation API, such as Google Translation API. To begin, we assign unique numbers to represent each concept in the ontology, ensuring that these numbers are mapped back to the original concepts. By doing so, we can translate these numbers into any desired language by invoking a custom-designed function that accepts a language code for the target language. The translated results can then be stored as attributes of nodes in the graph, reducing time complexity by avoiding repeated API calls for already translated language ontologies.

3.1.3 Updation of existing ontology

As the world around us is constantly transforming, there is a need for continuous updating in how we structurally represent knowledge. This is particularly relevant for existing ontologies, as new text corpora may contain information that current ontologies are unable to represent, causing them to fall short. Hence, ensuring that ontologies remain fresh and up-to-date becomes essential to better represent recent and new knowledge. In this context, it involves comparing the already existing ontology with the newly created one to see if there is something that can be added to the original one. This is done by taking the edges and documents stored in ArangoDB for both the vertices and performing the join operation to add the new nodes and edges in the existing ontology. Thereafter, a new updated ontology is formed. This join operation is performed using libraries like pandas and AQL queries to retrieve data in our case. For instance, let's consider the two graphs depicted in Fig. 3. and Fig. 4. Since they are not the same, the ontologies are compared, and a new graph, shown in Fig. 5, is formed, incorporating the existing ontologies. By employing graph and join queries in updating ontologies, we can ensure that these knowledge representations remain relevant, accurate, and comprehensive in the face of evolving information and text corpora.

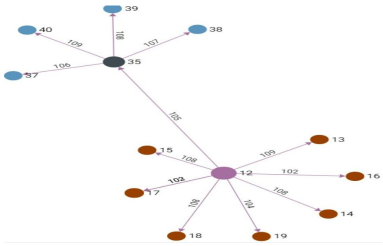

Fig. 6. Updated ontology with sample text 1 (existing ontology) and sample text 2 (new ontology)

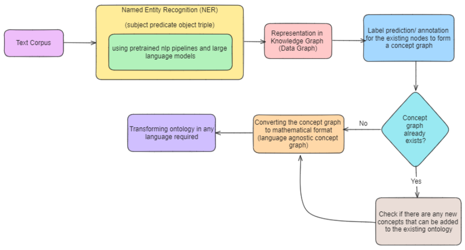

Fig. 7. Flow diagram of language agnostic ontology generation

3.2 Mathematical Formulation

We have modeled the ontologies as a set of triplets (subject, predicate, object). Let's say we already have ontology A in our system and we have learned about ontology B based on the new corpus, in this section we merge these two ontologies to create a final ontology. The merging is done using the ID-based approach depicted in Algorithm 1. This approach basically assigns a unique ID to concepts that exist in the ontology in turn generating a mathematically represented ontology.

The next step involves merging the ontologies represented in terms of IDs with is done as explained in Algorithm 2.

Ontology 1 (A):

A_1^s , A_1^P , $A_1^O \rightarrow A_1^I$
A_2^s , A_2^P , $A_2^O \rightarrow A_2^I$
. .
. .
A_n^s , A_n^P , $A_n^O \rightarrow A_n^I$

Where A_x^s, A_x^P, A_x^O and A_x^I represent the Subject, Predicate, Object and Instance (Triple) in Ontology A

Ontology 2 (B):

B_1^s , B_1^P , $B_1^O \rightarrow B_1^I$
B_2^s , B_2^P , $B_2^O \rightarrow B_2^I$
. .
. .
B_n^s , B_n^P , $B_n^O \rightarrow B_n^I$

Where B_x^s, B_x^P, B_x^O and B_x^I represent the Subject, Predicate, Object and Instance (Triple) in Ontology B

S_A = { A_1^I, A_2^I, ...A_n^I }
S_B = { B_1^I, B_2^I, ...B_n^I }

where S_A and S_B are set of instances from ontology A and B respectively.

Let the updated graph be C
C = $S_A \cup S_B$

Algorithm 1 Create ID based Ontology (Ontology O)

Initialize an empty Graph O_gen.
for each node n in graph O **do**
  $ID \leftarrow$ IdentityService(n) Call IdentityService to get the ID for the node.
  Add node with ID ID to graph O_gen.
end for
for each edge e in graph O **do**
  $ID \leftarrow$ IdentityService(e) Call IdentityService to get the ID for the edge.
  $source \leftarrow$ GetSourceNodeID(e) Get the ID of the source node for the edge.
  $target \leftarrow$ GetTargetNodeID(e) Get the ID of the target node for the edge.
  Add edge with ID ID, source $source$, and target $target$ to graph O_gen.
end for
Return Graph O_gen containing nodes and edges with their respective IDs.

Algorithm 2 Merge O_i and O_j

Initialize an empty map $visited$ to store if a predicate has already been added or not.
Initialize an empty Ontology/Graph O.
for each predicate p in graph O_i **do**
  Add predicate p to graph O.
  Mark predicate p as visited in map $visited$.
end for
for each predicate p in graph O_j **do**
  if p is not already present in map $visited$ **then**
    Add edge p to graph O. Mark predicate p as visited in map $visited$.
  end if
end for
Output: The graph O representing the merge of O_i and O_j.

4 RESULTS

The data-driven approach described above enables language agnostic ontology generation using deep learning models. The initial step involves utilizing the input text corpus as the foundation for building the ontology. By structuring the unstructured data through knowledge graph representations, the system extracts entities and relationships via natural language processing (NLP), specifically named entity recognition and dependency parsing. These extracted entities are then transformed into triples to be represented in a knowledge graph format, as illustrated in Fig. 1 and Fig. 2, showcasing two distinct text prompts in the knowledge graph format. To ensure the relevancy and factual accuracy of the knowledge graphs, a crucial subsequent step involves cleaning the graphs, eliminating unnecessary nodes and irrelevant information. Following the cleaning process, the nodes are categorized and annotated using annotation models, resulting in the formation of a final ontology. This entire process is automated, deriving the ontology directly from the unstructured data corpus, as depicted in Fig. 3 and Fig. 4.

Once the ontology is established, the next challenge is to update it with new information and extend its scope. For instance, consider Fig. 3. as the original ontology and Fig. 4. as the new ontology obtained. Since both ontologies have a common node 'company,' it implies the existence of fresh data about companies that can be incorporated into the existing ontology. To achieve this, the system verifies that the two ontology graphs are different. Subsequently, the system appends the extra nodes from the new ontology to the original one, resulting in an updated ontology (Fig. 5) that encompasses the newly acquired information. This newly acquired ontology is language agnostic, allowing it to be transformed into any required language using translation APIs. The function made for translating the ontology into different languages takes as the input the language name and the language code.

```
translate_ontology("English", "en")
translate_ontology("Japanese", "ja")
translate_ontology("tamil", "ta")
translate_ontology("hindi", "hi")
```

Fig. 8. Api function used to translate ontology to English, Hindi, Tamil, and Japanese

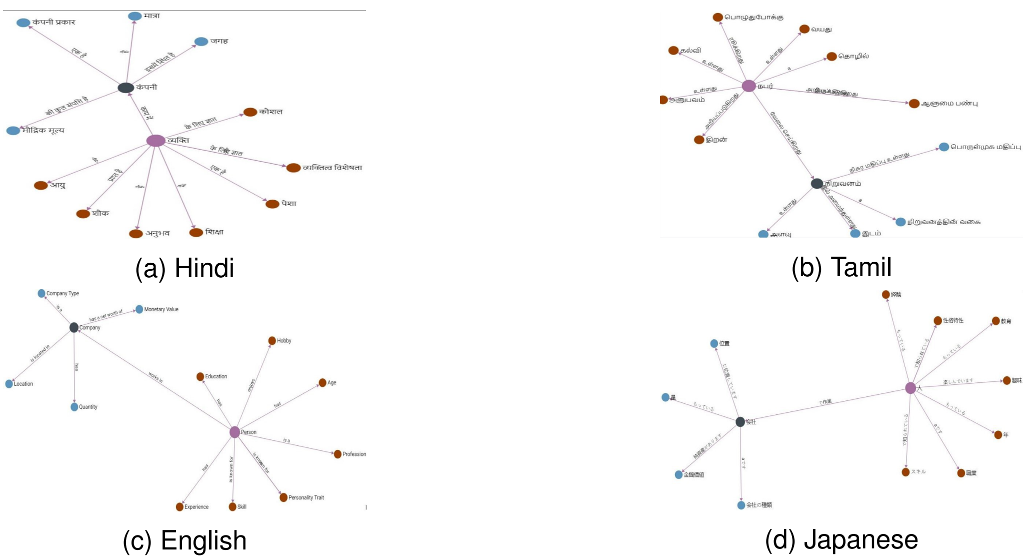

Fig. 9. Output depicting Language Agnostic domain-independent ontology

Fig. 9. displays the base ontology mathematically represented and translated into different languages such as Hindi, Tamil, English, and Japanese respectively. This is achieved by providing a mathematical representation for each concept in the original ontology, establishing a framework for generating a language agnostic ontology.

The research successfully addresses the need for an automated mechanism to generate ontologies from unstructured text data without requiring human intervention. By leveraging deep learning models, and NLP techniques, the system autonomously processes raw text data, creates knowledge graphs, annotates nodes, and generates a comprehensive ontology that is language agnostic. Moreover, the system's ability to accommodate new information enhances the ontology's relevance and applicability over time.

5 CONCLUSION

In conclusion, the paper aims to automate the generation of language-agnostic ontologies from raw, unstructured text corpora and also update existing ontologies [3]. Ontologies serve as powerful schema frameworks, enabling efficient, uniform, and comprehensive knowledge representation [1]. When utilized as blueprints for building knowledge graphs, they enhance the structure, semantics, and utility, making knowledge graphs potent tools for representation, data integration, reasoning, and exploration across diverse domains and applications. However, developing ontologies typically requires manual effort and a skilled team of experts with domain and ontology knowledge [6]. This challenge is addressed in the paper by proposing a data-driven approach to ontology generation [7]. Through natural language processing, the input text containing data is processed to extract concepts and create the ontology graph. This approach ensures robust, efficient, and accurate schema generation without human intervention [5]. Moreover, the created ontologies can be updated as new text prompts may introduce novel information not present in the existing ontology. This is done by various mechanisms such as graph queries and data manipulation libraries like pandas. To tackle the problem of ontologies being language-specific, mathematical representations of concepts were used to make the ontologies easy to transform into various languages using translation APIs [8]. This allowed the ontology to not only be domain-specific but also provided a way for them to be represented in a language-agnostic way. Overall, the automatic ontology generation reduces the need for manual curation, enhancing the overall robustness and efficiency of the process.

6 FUTURE WORK

In the future, research aims to process even more complex sentences as text input. These sentences may exhibit a richer, more human-like vocabulary with diverse and vast semantic relationships that current language models may not fully capture or comprehend. To achieve this, the focus lies on selecting models that can process and analyze sentences at a deeper level. These models will be trained on a massive corpus of diverse texts, enabling them to learn intricate relationships between words, grammar, semantics, and pragmatics within a language. This approach allows for scope in terms of validating and filtering sentences effectively. Furthermore, to enhance the accuracy of the approach, a mechanism for error control, thresholding, and validation will be implemented. Reference ontologies and knowledge graphs will play a crucial role in achieving this, as they will aid in cleaning the knowledge graphs and ontologies generated. By incorporating these enhancements, the research aims to create more accurate, precise, and versatile ontologies that are domain-independent and can be automatically generated.

COMPETING INTERESTS

Authors have declared that they have no known competing financial interests OR non-financial interests OR personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

[1] Elnagar S, Yoon V, Thomas MA. An automatic ontology generation framework with an organizational perspective. Proceedings of the 53rd Hawaii International Conference on System Sciences; 2020.
Retrieved from https://doi.org/10.48550/arXiv.2201.05910

[2] Osman AH, Barukub OM. Graph-based text representation and matching: A review of the state of the art and future challenges. in IEEE Access. 2020;8:87562-87583.
DOI: 10.1109/ACCESS.2020.2993191

[3] Shatalov O, Ryabova N. Named entity recognition problem for long entities in english texts. 2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT), LVIV, Ukraine. 2021;76-79.
DOI: 10.1109/CSIT52700.2021.9648768

[4] Ding, Ying, Schubert Foo. Ontology research and development. Part I - A review of ontology generation. Journal of Information Science. 2002;28.4:123-136.

[5] Lingfei Wu, Yu Chen, Kai Shen, Xiaojie Guo, Hanning Gao, Shucheng Li, Jian Pei, Bo Long. Graph neural networks for natural language processing: A survey; 2023.

[6] Jetschni J, Meister VG. Schema engineering for enterprise knowledge graphs: A reflecting survey and case study. 2017 Eighth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt. 2017;271-277.
DOI: 10.1109/INTELCIS.2017.8260074

[7] Ibrahim, Shimaa, et al. From monolingual to multilingual ontologies: The role of cross-lingual ontology enrichment. Semantic Systems. The Power of AI and Knowledge Graphs. Springer, Cham. 2019;215-230.

[8] Gu G, Nam Y, Park K, Galil Z, Italiano GF, Han WS. Efficient graph isomorphism query processing using degree sequences and color-label distributions. 2022 IEEE 38th International Conference on Data Engineering (ICDE), Kuala Lumpur, Malaysia. 2022;872-884.
DOI: 10.1109/ICDE53745.2022.00070

[9] Chen Y, Xing X. Constructing dynamic knowledge graph based on ontology modeling and neo4j graph database. 2022 5th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China. 2022;522-525. DOI: 10.1109/ICAIBD55127.2022.9820199

Peer-review history:
The peer review history for this paper can be accessed here:
http://www.sdiarticle5.com/review-history/109404